# Detection of Musical Event Drop from Crowdsourced Annotations Using a Noisy Channel Model

Naveen Kumar, Shrikanth S. Narayanan
komathnk@usc.edu, shri@sipi.usc.edu
*Signal Analysis and Interpretation Lab (SAIL)*
*University of Southern California, Los Angeles*
http://sail.usc.edu

## ABSTRACT

This paper describes the algorithm for our submission to the MediaEval 2014 crowdsourcing challenge. We perform a Maximum Likelihood (ML) estimation of the true label, using only the multiple noisy labels. Each annotator's decision is modeled by a die-toss based on which the annotator changes the true label. We learn parameters of this noisy channel model using the Expectation-Maximization algorithm. We also show that using a smaller number of annotators in the model than the actual number can give better accuracy because there is more data per annotator to estimate the parameters reliably.

## 1. INTRODUCTION

The Mediaeval 2014 crowdsourcing challenge [3] involves multiple noisy annotations for presence of the musical event *drop* in 15s clips taken from Electronic Dance Music (EDM) genre music. Each annotator assigns one of 3 class labels depending on the extent to which the event is present in the 15 second clip. For each such clip atleast 3 unique annotations are available from different annotators. The total number of unique annotators is 30, however the bulk of annotations are done by a handful of them (Fig.1).

The typical approach to modeling multiple noisy annotations is to model each of the $M$ annotators as a noisy channel that distorts the true label $Y$ into a noisy annotation $\widetilde{Y}^k$ for each of the $K$ annotations, $k = 1 \ldots, K$ per song. This can either be done in a data-independent [4] or a data-dependent fashion [1]. However, in the current problem at hand, these methods are not readily applicable, beause of our lack understanding of good features for the task. The 2014 crowdsourcing challenge dataset comprises of only noisy annotations and without any ground truth the process of feature design is difficult.

Hence, we use a much simpler model instead, based on [2] which only uses the multiple noisy annotations and models each annotator as a noisy channel that corrupts the "true label" ($Y$) by tossing a $B$-faced die where $B$ is the number of classes and the die is chosen depending on $Y$.

## 2. NOISY CHANNEL MODEL

For $N$ songs in the dataset, we denote the $k^{th}$ annotation for the $i^{th}$ song as $\widetilde{Y}_i^k$. The number of annotations can vary for each song. In addition, we denote the annotator id for each annotation $\widetilde{Y}_i^k$ using $A_i^k$. This information is provided on our dataset. For each annotator $m$ we denote the parameters of her noisy channel model by $\Lambda^m$. As an example if we denote $p_{ik} = Pr(Y_i = k)$ and $q_{ij} = Pr(\widetilde{Y}_i = j)$ then the $m^{th}$ annotator distorts her label as $q_i = \Lambda^m p_i$.

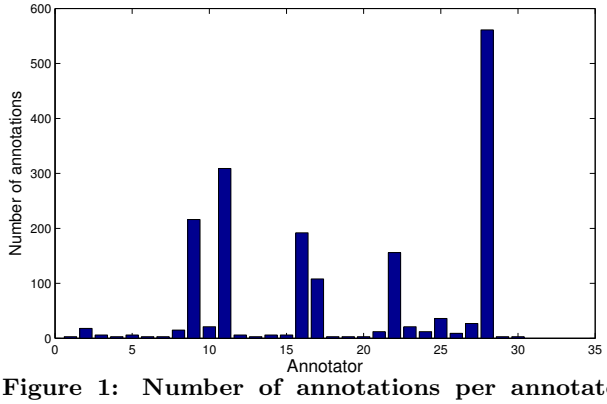


**Figure 1: Number of annotations per annotator. Note that most of the annotations are from a few annotators.**

We treat the true label $Y_i$ as a hidden parameter and perform Expectation Maximization to estimate it for each parameter, learning the model parameters $\Lambda^m$ at the same time. Since the annotator ids for each annotation are known to us, it is straightforward to compute the total data likelihood. For a dataset $\mathcal{D} = \{Y, \widetilde{Y}, A\}$ it is shown in Eqn.(1).

$$Pr(\mathcal{D}; \Lambda^{1\cdots M}) = \prod_{i=1}^{N} Pr(Y_i) \prod_k P(\widetilde{Y}_i^k, A_i^k | Y_i; \Lambda^{1\cdots M}) \quad (1)$$

$$= \prod_{i=1}^{N} Pr(Y_i) \prod_k \Lambda_{ab}^m; \text{ if } A_i^k = m, \widetilde{Y}_i^k = a, Y_i = b \quad (2)$$

Note that since $Y_i$ is a latent variable, in practice we shall maximize a lower bound of this likelihood function by taking an expectation w.r.t posterior distribution of the latent variable. This amounts to replacing $b$ by a soft label $p_{ib}$,

$$\mathbb{E}[Pr(\mathcal{D})] = \prod_{i=1}^{N} Pr(Y_i) \prod_k \prod_b (\Lambda_{ab}^m)^{p_{ib}}; \text{ if } A_i^k = m, \widetilde{Y}_i^k = a \quad (3)$$

where $p_{ib}$ is defined as earlier. We compute $p_{ib}$ formally in


*MediaEval 2014 Workshop,* October 16-17, 2014, Barcelona, Spain

the next section by estimating the posterior distribution of true labels given the noisy ones.

## 2.1 Expectation Step

In this step, we estimate the posterior probability of the latent variable $Y_i$ given the noisy annotations $\widetilde{Y}_i^k$,, model parameters $\Lambda^{1\ldots M}$ and class priors $\eta_b = Pr(Y_{ib} = 1)$. This is done as follows

$$Pr(Y_{ib} = 1|\widetilde{Y}_i^{1\ldots K}) = \frac{Pr(\widetilde{Y}_i^{1\ldots K}|Y_{ib} = 1)Pr(Y_{ib} = 1)}{\sum_j Pr(\widetilde{Y}_i^{1\ldots K}|Y_{ij} = 1)Pr(Y_{ij} = 1)} \tag{4}$$

We denote this as $\mu_{ib}$ and note this can be computed by knowledge of parameters $\Lambda^{1\ldots m}$ and $Pr(Y_{ib} = 1)$ that we shall refer to as $\eta_b$.

## 2.2 Maximization Step

This step performs optimization of the alternate likelihood function. The M-step in this case can be performed by simple count and divide as follows

$$\eta_b = \sum_{i=1}^{N} \mu_{ib}/N \tag{5}$$

$$\Lambda_{ab}^m = Pr(\widetilde{Y}_i^k = a|Y_i = b, A_i^k = m) \tag{6}$$

$$= \frac{\sum_{i=1}^N \sum_k \mu_{ib}\delta(\widetilde{Y}_i^k = a, A_i^k = m)}{\sum_{i=1}^N \sum_k \mu_{ib}\delta(A_i^k = m)} \tag{7}$$

To estimate the parameter $\Lambda_{ab}^m$ we count all probability mass for true class $b$ from annotator $m$ when the noisy label annotated was $a$. This is divided by the probability mass for annotator $m$, for the true label $b$ irrespective of the annotator's label.

We keep repeating these steps till the update in log-likelihood is below a certain threshold.

## 2.3 Uniqueness and Initialization

The EM algorithm can be shown to be a gradient ascent on log likelihood and hence is prone to getting stuck in local optima. Moreover, for this specific model there is an inherent non-uniqueness resulting from assignment of class labels. This means that by changing the order of columns in the parameters $\Lambda^m$ we can obtain a different permutation of true class labels. Each such permutation will still yield the same value of log-likelihood and is hence an equally optimal solution. This makes a good initialization of the EM algorithm important in this case. We use labels obtained using a majority vote as the initial estimates for $\mu_{ib}$.

Additionally, as pointed earlier most of the annotations are from a handful of annotators. This can lead to poor parameter estimates for annotators with few annotations. Thus, we choose the number of annotators $M$ within the model to be smaller than the actual number of annotators. We use $M = 8$ for the submitted runs based on a rough estimate from Fig.1. The annotator ids for top $(M - 1)$ annotators by number of annotations are retained. The rest are grouped together under the $M^{th}$ annotator id. The effect of varying the parameter $M$ is shown in Fig.2. Finally, to deal with numerical instabilities resulting from dividing by small numbers in the M-step we use Laplace smoothing.

**Table 1: The two systems submitted to the challenge with different initializations. Results show unweighted (UWF) and weighted F1-score (WF).**

| System | WF | UWF | Submitted |
|---|---|---|---|
| majority vote | 0.68 | 0.59 | |
| EM-random-init | 0.16 | 0.23 | ✓ |
| EM-majority-init | 0.73 | 0.65 | ✓ |

# 3. RESULTS AND CONCLUSIONS

We submitted two systems using the proposed method using a random initialization and the other using majority voted labels. The results are shown in Table 1. We compare the results against a high-fidelity annotation that is assumed to be the ground truth for the purposes of this challenge.

The accuracy for the submitted systems indicate that a proper initialization of the EM-algorithm is important as anticipated. Using labels obtained through majority voting of multiple noisy annotations allows us to obtain better results compared to simple sample level majority voting. Results are also sensitive to the number of annotators selected, and in the future we would like to automatically learn the paramter $M$.

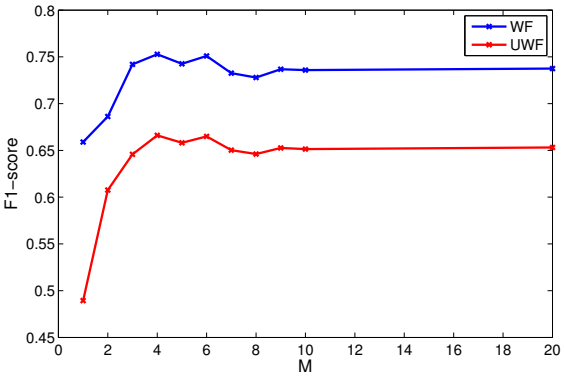


**Figure 2: Figure shows the effect of the assumed number of annotators $M$ on system F1-score.**

# 4. REFERENCES


[1] K. Audhkhasi and S. S. Narayanan. Data-dependent evaluator modeling and its application to emotional valence classification from speech. In *INTERSPEECH*, pages 2366–2369, 2010.

[2] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[3] M. L. Karthik Yadati, Pavala S.N. Chandrasekaran Ayyanathan. Crowdsorting timed comments about music: Foundations for a new crowdsourcing task. In *MediaEval 2014 Workshop*, Barcelona, Spain, October 16-17 2014.

[4] V. C. Raykar, S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual international conference on machine learning*, pages 889–896. ACM, 2009.